

Rotación de Objetos Tridimensionales Alrededor de una Recta Implementación en MATHEMATICA

Walter Mora F.

Instituto Tecnológico de Costa Rica

Escuela de Matemática

Resumen.

Si A es una matriz ortogonal, Av es una rotación del vector v alrededor de un vector u . En este trabajo, a partir de un ángulo θ y un eje de rotación u arbitrario, se obtiene una matriz ortogonal que describe esta rotación. Se hace una implementación en MATHEMATICA y se aplica a algunos objetos tridimensionales (se incluyen animaciones 3D). También se discute la relación entre los ángulos de Euler y el ángulo y el eje de rotación.

Introducción

Cuando se manipulan gráficos tridimensionales por computadora (en animación y simulación, etc.), una tarea muy conveniente es hacer rotaciones de estos objetos alrededor de una recta en el espacio. Cuando el objeto tridimensional es una lista de primitivas y las primitivas son una lista de puntos $P = \{P_1, \dots, P_k\}$ entonces, para hacer una animación del objeto tridimensional, que incluya rotaciones, lo mejor es tener una parametrización $P_i = P_i(\theta)$ del objeto. La animación es una sucesión de gráficos (frames) obtenidos al hacer variar θ . Una parametrización de los puntos, para hacer una rotación, se puede lograr con herramientas matriciales (matrices ortogonales). Una manipulación más poderosa de las rotaciones se puede lograr con el álgebra de cuaterniones, con esta última se puede establecer, de manera sencilla, la relación entre la rotación alrededor de un eje y los ángulos de Euler. En las bibliotecas estándar de MATHEMATICA 3.0 (4.0) hay dos alternativas para rotar un objeto tridimensional, una es usando ángulos de Euler (`RotateShape[]`) y otra es manipulando el `ViewPoint` (`SpinShow[]`). Ambas opciones son en general inadecuadas para tareas generales de animación. En el caso de los ángulos de Euler, no hay una relación adecuada entre estos ángulos y un eje de giro arbitrario. Sin embargo, se puede implementar un comando (de manera relativamente sencilla) para la rotación de cualquier objeto tipo `Graphics3D` alrededor de un eje arbitrario.

Matrices de rotación

Si A es ortogonal (i.e. $A^t A = I$) y si $\theta = 1$ es un valor propio de A y u es un vector propio asociado a $\theta = 1$, entonces $\langle Av, u \rangle = \langle Av, Au \rangle = \langle v, u \rangle$. Esto hace que Av y v tengan la misma norma y el mismo ángulo respecto a u . Av es una rotación de v alrededor de u .

En general, si A es ortogonal, Av es una rotación de v (algunas de estas rotaciones son 'impropias', es decir, corresponden a reflexiones, inversiones o 'rotoreflecciones'). Ahora, dada A ortogonal, cuál es el ángulo de giro y cuál es el eje de giro?

TEOREMA (Matrices ortogonales)

Sea $A_{3 \times 3} = (a_{ij})$ real ortogonal

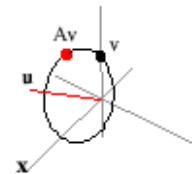
Sean ω y $\bar{\omega}$ complejos conjugados. Si $\lambda_1 = 1$, $\lambda_2 = \omega$ y $\lambda_3 = \bar{\omega}$ son los valores propios de la matriz A , entonces Av es una rotación de v , de ángulo ϕ , alrededor de la recta generada por cualquier vector propio u asociado al valor propio $\theta = 1$. El ángulo de rotación ϕ satisface la ecuación $\cos(\phi) = \frac{1}{2}(\text{Tr}A - 1)$. Además podemos tomar

$$u = \left(\frac{a_{32} - a_{23}}{2\text{sen}\phi}, \frac{a_{13} - a_{31}}{2\text{sen}\phi}, \frac{a_{21} - a_{12}}{2\text{sen}\phi} \right)$$

- En la fórmula del teorema si $\phi = 0$, u es indeterminado. Esto se considera razonable puesto que el eje de rotación pierde sentido en este caso.

EJEMPLO 1

La matriz ortogonal $A = \begin{pmatrix} \frac{\sqrt{2}}{3} & -\frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \end{pmatrix}$



cumple las condiciones del teorema, sus valores propios son $\theta = 1$, $\theta = 0.550477 \pm 0.83485 i$. El ángulo de rotación es 1.31812 radianes alrededor de $u = (0.663291, -0.210819, 0.508961)$.

Parametrización para la rotación alrededor de una recta

Consideramos primero el problema de encontrar una matriz de rotación correspondiente a un giro de ángulo $\frac{1}{2}$ alrededor del vector unitario u . Este problema se puede resolver geoméricamente

TEOREMA. Dado $\frac{1}{2}$ y u unitario entonces

$$Av = v + (\text{sen}\phi)(u \times v) + 2(\text{sen}^2 \frac{\phi}{2})u \times (u \times v)$$

es un giro de v de ángulo $\frac{1}{2}$ alrededor del vector u .

Demostración: ver [2].

Implementación en MATHEMATICA

La implementación en MATHEMATICA requiere de dos ingredientes, la fórmula de rotación y un procedimiento para aplicar la fórmula a una imagen tridimensional de tipo Graphics3D[]. La idea es simple: navegamos en el objeto tridimensional buscando las primitivas con "Head" Line, Polygon o Point y aplicamos la fórmula de rotación a la lista de puntos de cada primitiva. En el módulo (programa) ortonormalizamos u por si se le olvida al usuario.

| Giro alrededor de un vector |
|--|
| <pre> Rotar[objeto_, u_, phi_] := Module[{A}, A = (# + Sin[phi]*Cross[u/Sqrt[u.u], #] + 2(Sin[0.5*phi]^2)*Cross[u/Sqrt[u.u], Cross[u/Sqrt[u.u], #]])&; objeto /. {poly : Polygon[_] => Map[A, poly, {2}], line : Line[_] => Map[A, line, {2}], point : Point[_] => Map[A, point, {1}]}]; </pre> |

Se puede usar la parametrización dada por el teorema anterior para rotar un vector alrededor de una recta que no pasa necesariamente por el origen: primero se traslada la recta hasta que pase por el origen, se hace la rotación y de nuevo se traslada a la posición original.

TEOREMA. Sea u un vector unitario. Consideremos la recta $L : Q + t u$. Sea $A v$ una parametrización correspondiente a un giro de ángulo $\frac{1}{2}$ alrededor del vector unitario u , entonces $A(v - Q) + Q$ es una rotación de v , de ángulo $\frac{1}{2}$, alrededor de la recta L .

Implementación en Mathematica

Procedemos de manera análoga que en la sección anterior

```

RotarLin[objeto_, Q_, u_, phi_] := Module[{A},
  A = ((# - Q) + Sin[phi]*Cross[u/Sqrt[u.u], (# - Q)] +
    2(Sin[0.5*phi]^2)*Cross[u/Sqrt[u.u], Cross[u/Sqrt[u.u], (# - Q)]] + Q)&;
  objeto /. {poly : Polygon[_] => Map[A, poly, {2}],
            line : Line[_] => Map[A, line, {2}],
            point : Point[_] => Map[A, point, {1}]}];

```

EJEMPLO 2 : Animación: rotación de una de las caras de un dodecaedro.

En MATHEMATICA, un dodecaedro es una lista de 12 polígonos {Polygon[{...}], ..., Polygon[{...]}. Cada comando Polygon[{V1, ..., V5}] contiene los 5 vértices de la cara correspondiente. Para hacer una rotación de una cara del dodecaedro, fijamos un vértice (digamos V1) y tomamos el punto medio del lado opuesto al vértice. La recta que une estos puntos será el eje de giro. Luego usamos el comando RotarLin[] definido anteriormente. El código de la animación sería el siguiente

```

<< Graphics`Polyhedra`
Clear[d, cara, u, Q];
d = Dodecahedron[];
cara = Graphics3D[{d[[1]]}];
u = d[[1,1,1]] - (d[[1,1,3]] + d[[1,1,4]])/2;

```

```

Q = d[[1, 1, 1]];
dode = Graphics3D[{ Drop[d, {1}] }];
RotarLin[ob_,Q_,u_,phi_]:=Module[{A},...];

Do[
Show[{dode, RotarLin[ cara, Q, u, t ]},
PlotRange->{{-1.5,1.5},{-1.5,1.5},{-1.5,1.5}},
AspectRatio -> Automatic,
ViewPoint -> {0.031, 0.032, 3.383},
Boxed -> False, Axes -> False]
, {t, 0, 2Pi, 0.2}]

```



A esta animación se le aplicaron efectos especiales en Microsoft Image Composer, aquí mismo se generó la animación "gif" para el Web con GifAnimator.

Parametrización de un círculo en un plano

El último teorema nos dice también, cómo parametrizar un circunferencia en un plano: consideremos el plano $\Pi : [(x, y, z) - O_c] \cdot \mathbf{n} = 0$ y la circunferencia C , sobre el plano Π , de centro O_c y radio $\|O_c - P\|$, $P \in C$. Una parametrización de C es $A(P - O_c) + O_c$.

EJEMPLO 4 : Circunferencia en un plano.

Considere el triángulo con vértices

$$A = (-1, 1, 0), B = (1, 1, 0), \text{ y } C = (0, 0, 0)$$

Podemos dibujar las bisectrices, el incentro y la circunferencia inscrita del triángulo. El incentro es la intersección de las bisectrices. Sean $V = A - C$, $W = B - C$ y $U = A - B$ entonces dos bisectrices son $L_1 : C + t \left(V + \frac{\|V\|}{\|W\|} W \right)$ $L_2 : B + s \left(U - \frac{\|U\|}{\|W\|} W \right)$ y el incentro es $I_c = L_1 \cap L_2 = \{(0, 2/3, 1/6)\}$.

Para parametrizar la circunferencia, tomamos como centro el incentro I_c ; el radio es $\|I_c - P\|$ donde $P = \text{Proy}_{(A - B)}(I_c - B)$, P es el punto de tangencia del círculo con un lado del triángulo. Parte del código para desplegar la figura es

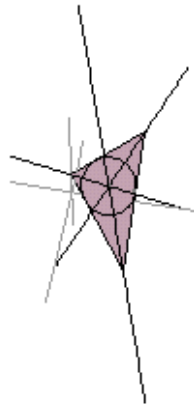
```

triangulo=Graphics3D[...];
Clear[A]; u=Cross[{-1,1,0},{1,1,-1}];
A[v_,Q_,u_,phi_]:=
(v-Q)+Sin[phi]*Cross[u/Sqrt[u.u],(v-Q)]+
2(Sin[0.5*phi]^2)*Cross[u/Sqrt[u.u],Cross[u/Sqrt[u.u],(v-Q)]]+Q;

circunferencia=
ParametricPlot3D[Evaluate[A[P,Ic,u,phi] /. phi->t]
,{t,0,2Pi},DisplayFunction->Identity];

```

```
Show[{triangulo, circunferencia},
      Boxed->False, Axes->False,
      ViewPoint->{0.405,2.768,1.904},
      DisplayFunction->$DisplayFunction];
```



Cuaterniones y Ángulos de Euler

En MATHEMATICA, la rotación de un objeto tridimensional se puede hacer con `RotateShape[obj, α, θ, ψ]` con $\theta \in [0, \pi]$. La rotación dada por los ángulos de Euler α, θ y ψ puede ser descompuesta en una sucesión de tres rotaciones sucesivas. La primera rota un ángulo α alrededor del eje Z, la segunda rota un ángulo θ alrededor del eje X y la tercera rota un ángulo ψ alrededor del eje Z nuevamente. `RotateShape[]` determina la matriz de rotación correspondiente y la aplica a todos los puntos del objeto geométrico. El comando `RotationMatrix3D[α, θ, ψ]` da la matriz de rotación correspondiente a las tres rotaciones sucesivas determinadas por los ángulos de Euler.

El problema es: dado $\frac{1}{2}$ y \mathbf{u} , cuál es la relación entre los ángulos de Euler y $\frac{1}{2}$ y \mathbf{u} ? La relación más sencilla se obtiene usando cuaterniones.

Los cuaterniones son objetos $[a, \mathbf{v}]$, donde a es un escalar y $\mathbf{v} \in \mathbb{R}^3$.

⊂ La suma y la multiplicación (no conmutativa) de cuaterniones se define así:

- $[a_1, \mathbf{v}] + [a_2, \mathbf{u}] = [a_1 + a_2, \mathbf{v} + \mathbf{u}]$
- $[a_1, \mathbf{v}][a_2, \mathbf{u}] = [a_1 a_2 - \mathbf{v} \cdot \mathbf{u}, a_1 \mathbf{u} + a_2 \mathbf{v} + \mathbf{u} \times \mathbf{v}]$
- La norma de un quaternion se define como $\|[a, \mathbf{v}]\| = \sqrt{a^2 + \mathbf{v} \cdot \mathbf{v}}$
- Si $\mathbf{1} = [1, (\mathbf{0}, \mathbf{0}, \mathbf{0})]$ entonces $[a, \mathbf{v}]\mathbf{1} = \mathbf{1}[a, \mathbf{v}]$.

- $$[\mathbf{a}, \mathbf{v}]^{-1} = \frac{[\mathbf{a}, -\mathbf{v}]}{\|[\mathbf{a}, \mathbf{v}]\|^2}, \text{ i.e. } [\mathbf{a}, \mathbf{v}][\mathbf{a}, \mathbf{v}]^{-1} = \mathbf{11}$$

Con esta álgebra se puede mostrar que un vector \mathbf{v} puede ser rotado alrededor de un eje aplicando la operación $\mathbf{q}\mathbf{v}\mathbf{q}^{-1}$. Lo que hacemos es, abusando del lenguaje, identificar \mathbf{v} con $[\mathbf{0}, \mathbf{v}]$. De esta manera, si $\mathbf{u} = \{u_1, u_2, u_3\}$ (unitario), si $\mathbf{q} = [\cos(\frac{1}{2}\theta), \text{sen}(\frac{1}{2}\theta)\mathbf{u}]$ y si $[\mathbf{0}, \mathbf{v}'] = \mathbf{q} [\mathbf{0}, \mathbf{v}] \mathbf{q}^{-1}$, se tiene que \mathbf{v}' es una rotación de \mathbf{v} , de ángulo $\frac{1}{2}\theta$, alrededor de \mathbf{u} .

Con este último resultado podemos determinar una relación entre los ángulos de Euler y una rotación de ángulo $\frac{1}{2}\theta$, alrededor de \mathbf{u} . Para esto, ponemos las rotaciones correspondientes a los ángulos de Euler en lenguaje de cuaterniones e igualamos:

$$[\cos(\frac{1}{2}\theta), \text{sen}(\frac{1}{2}\theta)\mathbf{u}] = [\cos(\frac{\alpha}{2}), \text{sen}(\frac{\alpha}{2})\mathbf{k}] [\cos(\frac{\theta}{2}), \text{sen}(\frac{\theta}{2})\mathbf{j}] [\cos(\frac{\psi}{2}), \text{sen}(\frac{\psi}{2})\mathbf{k}]$$

luego, desarrollando la multiplicación de la derecha e igualando, se obtiene

$$\begin{aligned} \cos(\frac{1}{2}\theta) &= \cos(\frac{\theta}{2}) \cos(\frac{\alpha + \psi}{2}) \\ u_1 &= -(\text{sen}(\frac{1}{2}\theta))^{-1} \text{sen}(\frac{\theta}{2}) \text{sen}(\frac{\alpha - \psi}{2}) \\ u_2 &= (\text{sen}(\frac{1}{2}\theta))^{-1} \text{sen}(\frac{\theta}{2}) \cos(\frac{\alpha - \psi}{2}) \\ u_3 &= (\text{sen}(\frac{1}{2}\theta))^{-1} \cos(\frac{\theta}{2}) \text{sen}(\frac{\alpha + \psi}{2}) \end{aligned}$$

Esto nos dice que, en general, para hacer animaciones que involucran rotaciones de objetos tridimensionales alrededor de una recta, es mejor usar una implementación matricial (o en términos de cuaterniones).

Bibliografía

- Adams A; Rogers, D. ``*Mathematical Elements for Computer Graphics*``. McGraw-Hill, NY, 1990.
- Altmann, S. ``*Rotations, Quaternions and Double Groups*``. Clarendon Press, Oxford. 1986.
- Noble, B; Daniel, J. ``*Algebra Lineal Aplicada*``. Prentice-Hall. 1989.
- Vince, J ``*3-D Computer Animation*``. Addison-Wesley, 1992.
- Wolfram, S. ``*The Mathematica Book*``. 4th ed. Wolfram Media/Cambridge University Press, 1999